In [3]:

```
# UNSUPERVISED MACHINE LEARNING ALGORITHEM (K-Means) on IRIS DATASET

#importing the required libraries
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd
```

In [5]:

```
#loading the required data
Data=pd.read_csv("C:/Users/sampada/Downloads/Iris.csv")
print(Data.head())
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
```

In [6]:

```
#let the no of clusters be 3
km=KMeans(n_clusters=3)
```

In [7]:

```
#building a model
model=km.fit(Data[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']])
```

In [11]:

```
model
```

Out[11]:

```
KMeans(n_clusters=3)
```

In [9]:

```
model.labels_
```

Out[9]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [8]:

```
#crosstab to verify the predicted values with original values of species
pd.crosstab(Data.Species,model.labels_)
```

Out[8]:

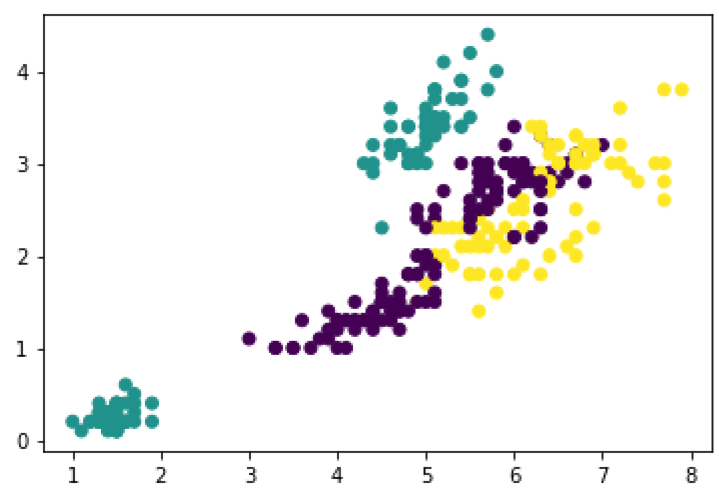
col_0	0	1	2
Species			
Iris-setosa	0	50	0
Iris-versicolor	48	0	2
Iris-virginica	14	0	36

In [ ]:

```
# we see that all Iris-Setosa have been correctly classified into one cluster named 1
#Iris Versicolor species(named as 0) has 2 records being missclassified as Iris virginica sp
ecies(named as 2)
#Iris verginica(2) has 14 records missclassified as iris-versicolor species(named as 0)
```

In [15]:

```
#visualizing the clusters
plt.scatter(Data.PetalLengthCm, Data.PetalWidthCm, c=model.labels_)
plt.scatter(Data.SepalLengthCm, Data.SepalWidthCm, c=model.labels_)
plt.show()
```



In [17]:

```
model_predict=km.predict(Data[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'
]])
```

In [18]:

```
model_predict
```

Out[18]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [19]:

```
km.cluster_centers_
```

Out[19]:

```
array([[5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [5.006     , 3.418     , 1.464     , 0.244     ],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

In [20]:

```
Data['predicted_KMeans_labels']=model.labels_
```

In [21]:

```
Data
```

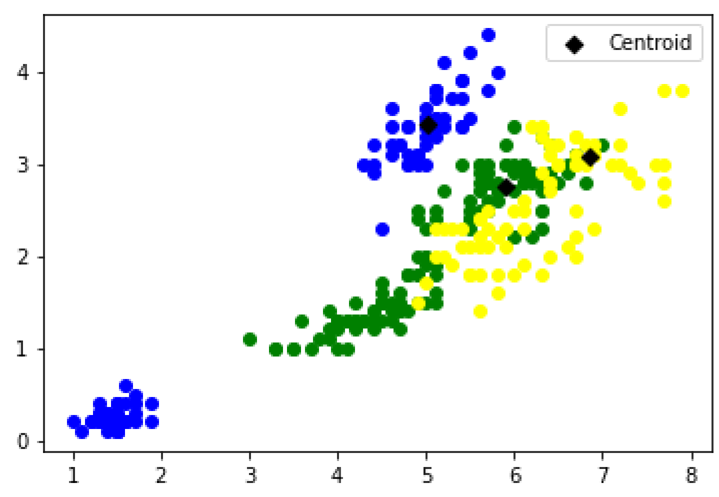
Out[21]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	predicted_KMeans_labels
0	1	5.1	3.5	1.4	0.2	Iris-setosa	1
1	2	4.9	3.0	1.4	0.2	Iris-setosa	1
2	3	4.7	3.2	1.3	0.2	Iris-setosa	1
3	4	4.6	3.1	1.5	0.2	Iris-setosa	1
4	5	5.0	3.6	1.4	0.2	Iris-setosa	1
...	...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica	2
146	147	6.3	2.5	5.0	1.9	Iris-virginica	0
147	148	6.5	3.0	5.2	2.0	Iris-virginica	2
148	149	6.2	3.4	5.4	2.3	Iris-virginica	2
149	150	5.9	3.0	5.1	1.8	Iris-virginica	0

150 rows × 7 columns

In [24]:

```
cl1=Data[Data.predicted_KMeans_labels==0]
cl2=Data[Data.predicted_KMeans_labels==1]
cl3=Data[Data.predicted_KMeans_labels==2]
plt.scatter(cl1.PetalLengthCm,cl1.PetalWidthCm, color='green')
plt.scatter(cl1.SepalLengthCm,cl1.SepalWidthCm, color='green')
plt.scatter(cl2.PetalLengthCm,cl2.PetalWidthCm, color='blue')
plt.scatter(cl2.SepalLengthCm,cl2.SepalWidthCm, color='blue')
plt.scatter(cl3.PetalLengthCm,cl3.PetalWidthCm, color='yellow')
plt.scatter(cl3.SepalLengthCm,cl3.SepalWidthCm, color='yellow')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1], color='black',marker="D",labe
l="Centroid")
plt.legend()
plt.show()
```



In [25]:

```
#calculating SSE for a range of K values
k_range=range(1,10)
SSE = []
for k in k_range:
    km=KMeans(n_clusters=k)
    km.fit(Data[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']])
    SSE.append(km.inertia_)
```

In [26]:

```
SSE
```

Out[26]:

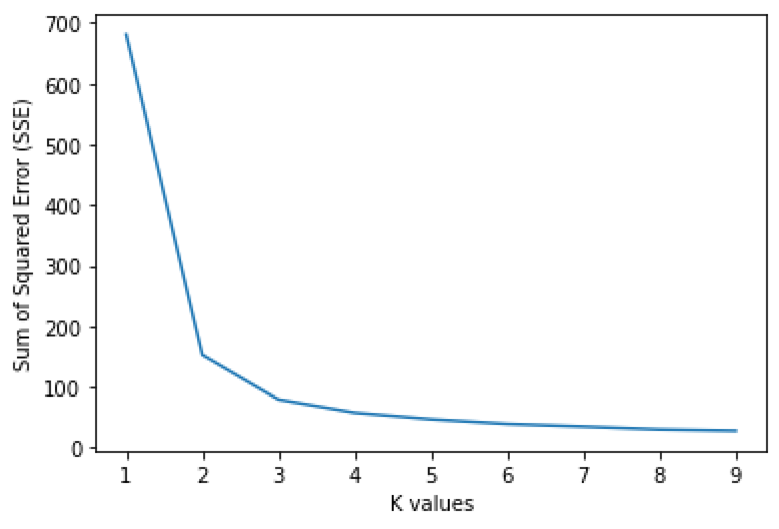
```
[680.8243999999996,
 152.36870647733915,
 78.94084142614601,
 57.317873214285726,
 46.56163015873017,
 38.93873974358975,
 34.62085338680927,
 29.955568877177583,
 27.76690692640694]
```

In [27]:

```
#plotting the elbow plot
plt.xlabel("K values")
plt.ylabel("Sum of Squared Error (SSE)")
plt.plot(k_range,SSE)
```

Out[27]:

```
[<matplotlib.lines.Line2D at 0x6ff5797520>]
```


Sum of Squared Error (SSE)
K values